Mithilesh Kabadi

Day 6 Practice Problems

Write a program that takes a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n.

```
read -p "Enter n: " n

for(( i=1; i<=n; i++ ))
do
	j=$((2**$i))
	echo -n "$j"
	printf "\n"
done
```

Write a program that takes an input and determines if the number is a prime.

```
echo -e "Enter Number : \c"
read n
for((i=2; i<=$n/2; i++))
do
  ans=$(( n%i ))
  if [ $ans -eq 0 ]
  then
    echo "Not a prime number."
    exit 0
  fi
done
echo "Prime number."
```

Write a program that computes a factorial of a number taken as input. 5 Factorial – 5! = 1 * 2 * 3 * 4 * 5

```
echo "Enter a number"
read num

fact=1

for((i=2;i<=num;i++))
{
  fact=$((fact * i))
}

echo $fact
```

Write a program to compute Factors of a number N using prime factorization method.

Write a program that takes a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n till 256 is reached

```
i=0
n=1
while [ $i -lt 256 ]
do
   i=$((2**$n))
   echo $i
   n=$(($n+1))
done
```

Write a Program where a gambler starts with Rs 100 and places Re 1 bet until he/she goes broke i.e. no more money to gamble or reaches the goal of Rs 200. Keep track of number of times won and number of bets made.

```
amt=100
count=0
while[ $amt!=0 ]
do
    while[ $amt<200 ]
    do
        read -p "Enter no between 1 to 10: " n
        side=$(($RANDOM%10 + 1));
        if[$n -eq $side]
        then
            amt=$(($amt+1))
            ((count++))
        else
            amt=$(($amt-1))
        fi
    done
done
echo "$count"
```

Extend the Flip Coin problem till either Heads or Tails wins 11 times

```
headCount=0
tailCount=0

for((i=0;i<=22;i++))
do
    side=$(($RANDOM%2 + 1));

    if(($side > 1))
    then
        while [ $headCount -lt 11 ]
        do
            headCount=$(($headCount+1))
        done
        else
                while [ $tailCount -lt 11 ]
                do
                        tailCount=$(($tailCount+1))
                done

        fi
done
echo "Head Count: $headCount"
echo "Tail Count: $tailCount"
```

Help user find degF or degC based on their Conversion Selection. Use Case Statement and ensure that the inputs are within the Freezing Point (0 °C / 32 °F) and the Boiling Point of Water (100 °C / 212 °F)

```
echo "a)degF"
echo "b)degC"
read -p "Enter your choice: " c

function degF(){
        read -p "Enter degC value: " degC
        value=$((degC*(9/5)+32))
        echo "$value"
        }

function degC(){
        read -p "Enter degF value: " degF
        value=$(((degF-32)*5/9))
        echo "$value"
        }

case $c in
        a)
         degF
         ;;
        b)
         degC
         ;;
        *)
         Default condition
```

```
    ;;
Esac
```

Write a function to check if the two numbers are Palindromes

```
function palindrome(){

    rev=""
    temp=$num

    while [ $num -gt 0 ]
    do

    s=$(( $num % 10 ))
    num=$(( $num / 10 ))


    rev=$( echo ${rev}${s} )
    done

    if [ $temp -eq $rev ];
    then
        echo "Number is palindrome"
    else
        echo "Number is NOT palindrome"
    fi
}

read -p "Enter your first number: " num
palindrome num
read -p "Enter your second number: " num
palindrome num
```

Take a number from user and check if the number is a Prime then show that its palindrome is also prime

```
function prime(){

    read -p "Enter your first number: " num1

    countPrime=0
    for((i=2; i<=$num1/2; i++))
    do
        ans=$(( num1%i ))
        if [ $ans -eq 0 ]
        then
            echo "$num1 is not a prime number."
            exit 0
        fi
    done
    echo "$num1 is a prime number."
    countPrime=$((countPrime+1))



}


function palindrome(){

    countP=0
    rev=""
    temp=$num

    while [ $num -gt 0 ]
    do

    s=$(( $num % 10 ))
    num=$(( $num / 10 ))


    rev=$( echo ${rev}${s} )
    done

    if [ $temp -eq $rev ]
    then
```

```
        echo "Number is palindrome"
        countP=$((countP+1))

    else
        echo "Number is NOT palindrome"
    fi
    prime

}

read -p "Enter your first number: " num
palindrome num
```